



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

object oriented DLL



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **object oriented DLL**Found **31,661** of **141,680**Sort results
by

relevance

[Save results to a Binder](#)[Try an Advanced Search](#)Display
results

expanded form

[Search Tips](#)[Try this search in The ACM Guide](#)
☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐1 [Object-oriented, single-source, on-line documents that update themselves](#)

Susan Korgen

 October 1996 **Proceedings of the 14th annual international conference on Systems
documentation: Marshaling new technological forces: building a
corporate, academic, and user-oriented triangle**

 Full text available: pdf(757.84 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)
2 [Object-oriented technology: Using SOM for tool integration](#)

Christina Lau

 October 1993 **Proceedings of the 1993 conference of the Centre for Advanced Studies
on Collaborative research: software engineering - Volume 1**

 Full text available: pdf(639.08 KB) Additional Information: [full citation](#), [abstract](#), [references](#)


IBM's System Object Model (SOM) is an object-oriented programming interface for building binary class libraries. When changes are made to a SOM class, client programs that use the SOM class will not need to be recompiled. The SOM Toolkit provides a set of frameworks for building object-oriented applications. These include Distributed SOM, the Persistence Framework, the Replication Framework, and the Emitter Framework. In this paper, we discuss our experience using SOM to handle the control aspect ...

3 [Type-Safe linking with recursive DLLs and shared libraries](#)

Dominic Duggan

 November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS),
Volume 24 Issue 6**

 Full text available: pdf(658.62 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



Component-based programming is an increasingly prevalent theme in software development, motivating the need for expressive and safe module interconnection languages. Dynamic linking is an important requirement for module interconnection languages, as exemplified by dynamic link libraries (DLLs) and class loaders in operating systems and Java, respectively. A semantics is given for a type-safe module interconnection language that supports shared libraries and dynamic linking, as well as circular ...

Keywords: Dynamic Linking, Module Interconnection Languages, Recursive Modules, Shared Libraries

4 Type-based hot swapping of running modules (extended abstract)

Dominic Duggan

October 2001 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN international conference on Functional programming**, Volume 36 Issue 10

Full text available:  pdf(150.34 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

While dynamic linking has become an integral part of the run-time execution of modern programming languages, there is increasing recognition of the need for support for hot swapping of running modules, particularly in long-lived server applications. The interesting challenge for such a facility is to allow the new module to change the types exported by the original module, while preserving type safety. This paper describes a type-based approach to hot swapping running modules. The approach is based on ...

Keywords: dynamic typing, hot swapping, module interconnection languages, shared libraries

5 The GMAP: a versatile tool for physical data independence

Odysseas G. Tsatalos, Marvin H. Solomon, Yannis E. Ioannidis

April 1996 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 5 Issue 2

Full text available:  pdf(228.04 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)


Physical data independence is touted as a central feature of modern database systems. It allows users to frame queries in terms of the logical structure of the data, letting a query processor automatically translate them into optimal plans that access physical storage structures. Both relational and object-oriented systems, however, force users to frame their queries in terms of a logical schema that is directly tied to physical structures. We present an approach that eliminates this dependence. ...

Keywords: Indexing, Materialized views, Physical data independence, Physical database design

6 An object-oriented modeling of the history of optimal retrievals

Yong Zhang, Vijay V. Raghavan, Jitender S. Deogun


September 1991 **Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval**

Full text available:  pdf(895.03 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

7 Recycling APL code into client/server applications

Richard J. Busman, Walter G. Fil, Andrei V. Kondrashev

June 1995 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on Applied programming languages**, Volume 25 Issue 4

Full text available:  pdf(994.90 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


This paper analyzes the current state of modern APL and deals with the authors' experiences in recycling and downsizing old APL applications and transforming them into more open information systems using client/server technology on different hardware and software platforms.

Keywords: APL/W, DB2, DDE, DLL, Dyalog, MS Windows, ODBC, SHARP APL, SQL, client/server, downsizing APL applications

8 Multilanguage programming with ada in the .Net environment

Jeffrey W. Humphries, Martin C. Carlisle, Terry A. Wilson

December 2003 **ACM SIGAda Ada Letters , Proceedings of the 2003 annual international conference on Ada: the engineering of correct and reliable software for real-time & distributed systems using ada and related technologies**, Volume XXIV Issue 1

Full text available:  [pdf\(207.95 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes our experiences in using Ada with other programming languages in the .NET environment. This paper explains our approach and presents lessons learned during our development of a real-world software project using .NET. We compare and contrast the languages used, justify our language choices, and present details of our efforts.

Keywords: A#, Ada 95, microsoft .NET environment, multilanguage programming

9 Release-to-release binary compatibility in SOM

Ira R. Forman, Michael H. Conner, Scott H. Danforth, Larry K. Raper

October 1995 **ACM SIGPLAN Notices , Proceedings of the tenth annual conference on Object-oriented programming systems, languages, and applications**, Volume 30 Issue 10


Full text available:  [pdf\(1.66 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

SOM (IBM's System Object Model) removes a major impediment to reuse in Object-Oriented Programming by facilitating the programming of release-to-release binary compatible class libraries. This is accomplished by supporting a large number of compatibility preserving transformations. Taken together these transformations compose a discipline for programming evolving class libraries.

10 Automatically connecting documentation to code with rose

Robert Pierce, Scott Tilley

October 2002 **Proceedings of the 20th annual international conference on Computer documentation**

Full text available:  [pdf\(256.32 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

One of the most common problems with program documentation is keeping it synchronized with the source code it purports to explain. One solution to this problem is to automate the documentation process using reverse engineering technology. Reverse engineering is an emerging branch of software engineering that focuses on recreating high-level information (such as program documentation) from low-level artifacts (such as source code). This paper describes an automated approach to maintaining the con ...

Keywords: application programming interface (API), automation, documentation, single sourcing, software engineering

11 Interaction points: exploiting operating system mechanisms for inter-component communications

Daniel G. Waddington, Ramesh Viswanathan

April 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue 2

Full text available:  [pdf\(1.08 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper introduces the concept of "interaction points" which are currently used in an experimental programming environment, the In-process Modular Programming (IMP) platform. IMP is a Microsoft COM-style platform designed specifically for building component-based applications in real-time and embedded environments, and used for research prototyping at Lucent Technologies, Bell-labs. Interaction points serve as a unified abstraction for a wide range of inter-component communication mechanisms. ...

Keywords: component engineering, embedded, interaction points, modular, network services, operating-system level communications, real-time

12 System applications and experience: A data mining based system supporting tactical decisions

Giuseppe Polese, Massimiliano Troiano, Genoveffa Tortora

July 2002 **Proceedings of the 14th international conference on Software engineering and knowledge engineering**

Full text available:  [pdf\(222.00 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)


We present a decision support system based on data mining algorithms to be used for tactical decisions. The system has been developed and engineered to solve a typical problem involving strategic decisions: supporting a trainer of a basketball team in making technical/tactical decisions. The experiments conducted on this application domain proved the effectiveness of the system and its underlying algorithms. The basketball domain stressed several aspects of decision making, proving that the used ...

Keywords: apriori algorithm, data mining, decision query, decision support systems, discovery model, verification model

13 Workflow enactment with continuation and future objects

Dragos A. Manolescu

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 37 Issue 11

Full text available:  [pdf\(322.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


An increasing number of software developers are turning to workflow to separate the logic and the control aspects in their applications, thus making them more amenable to change. However, in spite of recent efforts to standardize and provide reusable workflow components, many developers build their own. This is a challenging endeavor and involves solving problems which seem incompatible with the object paradigm and current object-oriented programming languages. In the context of an object-orient ...

Keywords: continuations, future objects, micro-workflow, trampolined style, workflow

14 Ensemble: a graphical user interface development system for the design and use of interactive toolkits

M. K. Powers

November 1989 **Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology**

Full text available:  [pdf\(1.67 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

User Interface Development Systems (UIDS), as opposed to User Interface Management Systems or UI Toolkits focus on supporting the design and implementation of the user interface. This paper describes Ensemble, an experimental UIDS that begins to explore the

electronic creation of interaction techniques as well as the corresponding design processes. Issues related to the impact on the components of the development system are discussed. Finally, problems with the current implementation and fu ...

15 Techniques and experiences with group support system implementation

Stephen C. Hayne, Mark Pendergast

October 1994 **Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  pdf(108.71 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes practical design and implementation techniques used for creating Group Support Systems (GSS) in networked PC environments. Examples of GSS based on the shared context model and implemented in Microsoft Windows are presented and experiences using 'C', 'C++', 'ObjectPascal' and 'Actor' languages included. Graphical user interfaces and multi-tasking extend traditional methods for supporting group work. The Network Based Object provides support for all inter and intra processor ...

Keywords: distributed computing, object oriented technology, software architecture

16 Models and techniques for the visualization of labeled discrete objects

Dinesh P. Mehta, Sartaj Sahni


March 1992 **Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing: technological challenges of the 1990's**

Full text available:  pdf(1.37 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 Weaving Ada 95 into the .net environment

Martin C. Carlisle, Ricky E. Sward, Jeffrey W. Humphries

December 2002 **Proceedings of the 2002 annual ACM SIGAda international conference on Ada: The engineering of correct and reliable software for real-time & distributed systems using Ada and related technologies**

Full text available:  pdf(254.58 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



This paper explains our efforts to add Ada to Microsoft's family of .NET languages. There are several advantages to weaving Ada into the Common Language Environment provided by the .NET environment. This paper explains our approach and current progress on the research. We provide the means to extract Ada specification files from Microsoft Intermediate Language (MSIL) code and compile Ada programs into MSIL.

Keywords: Ada 95, Microsoft .Net environment, common language runtime, just-in-time compiling

18 Execution characteristics of desktop applications on Windows NT

Dennis C. Lee, Patrick J. Crowley, Jean-Loup Baer, Thomas E. Anderson, Brian N. Bershad

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture**, Volume 26 Issue 3

Full text available:  pdf(1.42 MB)  [Publisher Site](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This paper examines the performance of desktop applications running on the Microsoft Windows NT operating system on Intel x86 processors, and contrasts these applications to the programs in the integer SPEC95 benchmark suite. We present measurements of basic instruction set and program characteristics, and detailed simulation results of the way these

programs use the memory system and processor branch architecture. We show that the desktop applications have similar characteristics to the integer ...

19 It knows what you're going to do: adding anticipation to a Quakebot

John E. Laird

May 2001 **Proceedings of the fifth international conference on Autonomous agents**

Full text available:  pdf(211.58 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The complexity of AI characters in computer games is continually improving; however they still fall short of human players. In this paper we describe an AI bot for the game Quake II that tries to incorporate some of the missing capabilities. This bot is distinguished by its ability to build its own map as it explores a level, use a wide variety of tactics based on its internal map, and in some cases, anticipate its opponents actions. The bot was developed in the Soar architecture and uses d ...

Keywords: anticipation, computer games, quake, soar

20 Microsoft Windows programming strategies

Mike Maxim

June 2000 **Crossroads**, Volume 6 Issue 4

Full text available:  html(23.57 KB) Additional Information: [full citation](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)